Predict, Prevent, and Evaluate: Disentangled Text-Driven Image Manipulation Empowered by Pre-Trained Vision-Language Model

Zipeng Xu[1]* Tianwei Lin[2] Hao Tang[3] Fu Li[2] Dongliang He[2]
Nicu Sebe[1] Radu Timofte[3] Luc Van Gool[3] Errui Ding[2]
[1]MHUG, University of Trento [2]VIS, Baidu Inc. [3]CVL, ETH Zürich
zipeng.xu@unitn.it

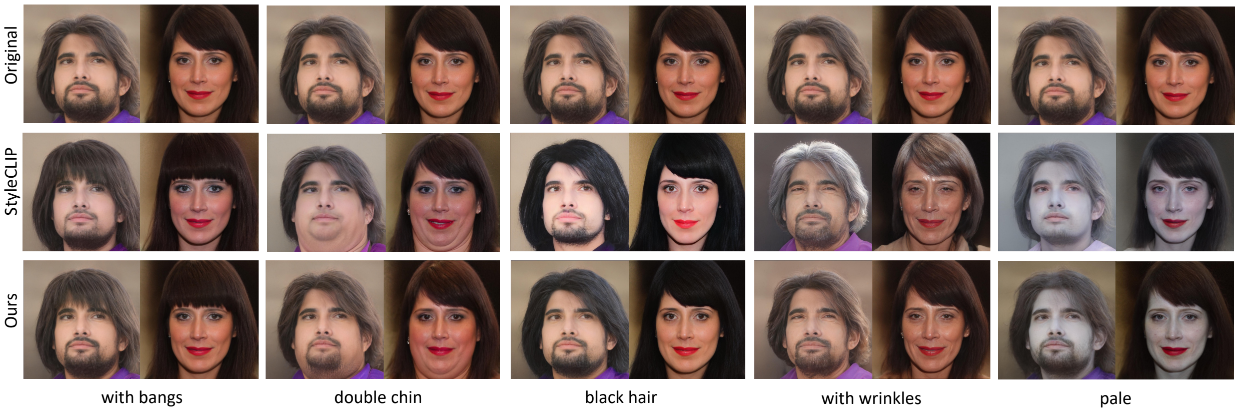


Figure 1. Comparisons on disentangled image manipulation between the StyleCLIP [31] baseline and our Predict, Prevent, and Evaluate (PPE). Ours manages to manipulate only the command-attribute (as indicated under each column) while remaining unchanged to the others.

Abstract

To achieve disentangled image manipulation, previous works depend heavily on manual annotation. Meanwhile, the available manipulations are limited to a pre-defined set the models were trained for. We propose a novel framework, i.e.*, Predict, Prevent, and Evaluate (PPE), for disentangled text-driven image manipulation that requires little manual annotation while being applicable to a wide variety of manipulations. Our method approaches the targets by deeply exploiting the power of the large-scale pre-trained vision-language model CLIP [32]. Concretely, we firstly* ***Predict*** *the possibly entangled attributes for a given text command. Then, based on the predicted attributes, we introduce an entanglement loss to* ***Prevent*** *entanglements during training. Finally, we propose a new evaluation metric to* ***Evaluate*** *the disentangled image manipulation. We verify the effectiveness of our method on the challenging face editing task. Extensive experiments show that the proposed PPE framework achieves much better quantitative and qualitative results than the up-to-date StyleCLIP [31] baseline. Code is available at https://github.com/zipengxuc/PPE.*

1. Introduction

Disentangled image manipulation [1, 8, 10, 12, 21, 23, 37, 38, 43, 44] aiming at changing the desired attributes of the image while keeping the others unchanged, has long been studied for its research significance and application value. Reaching this target is not easy, especially when attributes naturally entangle in the real world. Therefore, concrete attribute annotations are of vital importance, making disentangled image manipulation a labor-consuming task.

Several works [8, 10, 21, 23] use an encoder-decoder architecture and need manual annotations on multiple attributes of images. The models encode the original image and the manipulating attribute, then decode the manipulated image. Specifically, they use an attribute-specific loss to encourage the manipulation of a specific attribute while discouraging the others. The loss comes from pre-trained classifiers for all annotated attributes. Many recent works focus on latent space image manipulation since large-scale pre-trained GANs, *e.g.*, StyleGANs [15, 16], can generate

*The work was done during Zipeng Xu's internship at VIS, Baidu.

high-quality images from well-disentangled latent spaces. Despite the convenience of directly using the pre-trained GANs to generate images, all these methods need human annotations [1, 12, 37, 38, 43, 44]. Moreover, the available manipulating attributes are limited to the annotated set.

Recently, the rise of the large-scale pre-trained vision-language model CLIP [32] has brought a new insight. Since CLIP provides effective signals about the semantic similarity of image and text, various manipulations [11, 31, 36] can be performed with a text command and a CLIP-based loss, instead of exhaustive human annotations. Nevertheless, achieving disentangled image manipulation is still tricky. For instance, StyleCLIP [31] introduces three methods: latent optimization and latent mapper take no consideration of achieving disentangled results; global direction, which is based on the more disentangled $\mathcal{S}$ latent space [45], needs human trials-and-errors to find appropriate parameters in each case to reach the expected effects. To only manipulate a desired attribute, TediGAN [46, 47] merely revise the latent vectors of layers corresponding to that attribute. Yet, they have to figure out in advance the relations between attributes and layers in StyleGAN.

In this paper, we explore achieving disentangled image manipulation with as less human labor as possible. We propose a novel framework, *i.e.*, Predict, Prevent, and Evaluate (PPE), to approach the target by leveraging the power of CLIP in depth. Firstly, we propose to **Predict** the possibly entangled attributes for given text commands. We assume that the entanglements result from the distributions of attributes in the real world. Therefore, we draw support from CLIP to find the attributes that appear most frequently in the command-related images, then regard the attributes of high co-occurrence frequency as the possibly entangled attributes. Secondly, we introduce a novel entanglement loss to **Prevent** entanglements during training. The loss punishes the changes of the possibly entangled attributes before and after the manipulation, so as to enforce the model to find a less disentangled manipulating direction. Lastly, based on the predicted entangled attributes, we introduce a new evaluation metric to simultaneously **Evaluate** the manipulation effect and the entanglement condition. The manipulation effect is measured based on the change of command-attribute while the entanglement condition is based on the change of the entangled attributes, before and after manipulation. All the changes are estimated according to the CLIP distance between the texts of attributes and the images.

To evaluate, we implement our method based on the simple and versatile latent mapper from StyleCLIP and conduct experiments on the challenging face editing task, using the large-scale human face dataset CelebA-HQ [14, 25]. Qualitative and quantitative results indicate that we achieve superior disentangled performance compared to the StyleCLIP baseline. Meanwhile, we show that our results present a better linear consistency.

To conclude, our main contributions are as follows:

- We propose to predict entangled attributes for disentangled image manipulation.
- We propose a novel entanglement loss to prevent entangled manipulations during training.
- We propose a new evaluation metric that jointly measures the manipulation effect and the entanglement condition for disentangled image manipulation.
- By applying our method to the versatile StyleCLIP baseline, we manage to achieve disentangled image manipulation with very little manual labor. We conduct extensive experiments on the CelebA-HQ dataset and find that our qualitative and quantitative results are rather impressive.

2. Related Work

Disentangled Image Manipulation. Many works study learning a disentangled representation [3, 4, 9, 17, 21], so that disentangled image manipulation can be solved from the source. Due to the costly labor, a key challenge of such works is reducing the supervision for learning the desired disentanglement. Therefore, weakly-supervised and unsupervised methods have been explored [8,10,24,28]. Despite progress, all these methods are trained for a fixed set of attributes, thus supporting limited numbers of manipulations.

Recently, growing numbers of works focus on latent space image manipulation [12, 13, 37, 41, 45, 48] because of the remarkable large scale GANs like StyleGAN [15, 16], which can generate high-resolution images with well disentangled latent space. Thereby, these works firstly invert the image into the latent space through the GAN inversion method [16, 50] or an involved encoder [2, 29, 35], then accordingly compute the latent vector that can derive the manipulation result through the pre-trained large scale GANs. For each manipulating attribute, manual annotations are required, *e.g.*, on images [1, 37] and on unsupervisedly discovered directions in the latent space [12, 38, 43, 44].

Text-Driven Image Manipulation. There are studies that explore image manipulation with text commands as a guide. Some previous works [6, 18, 19, 27] use GAN-based encoder-decoder architectures, which encode the original image and text command, disentangle the semantics of the two modalities and decode the manipulated image. Instead of training a generator individually, the recent TediGAN [46, 47] and StyleCLIP [31] use pre-trained StyleGAN to generate images from manipulated latent vectors. To reach disentangled manipulation, TediGAN pre-defines an attribute-to-layer map and only changes the attribute-corresponding layers in StyleGAN. Besides, TediGAN conducts instance-level manipulation, which means the model is only applicable to one image that the model was optimized for. The latent mapper method in StyleCLIP is more general as the trained model can be applied to manipulate any in-domain image, but the results are usually entangled. The global direction method in StyleCLIP can realize a disentangled manipulation, but it requires manual trials-and-errors to find appropriate thresholds in the method. Our method proposes to achieve disentangled image manipulations with less manual effort by deeply leveraging the power from large scale pre-trained models. More than StyleCLIP, which merely minimizes the CLIP distances between command texts and manipulated images, we propose to predict, prevent, and evaluate entanglements via CLIP.

Large Scale Vision-Language Models. Following the success of large scale pre-trained language models, *e.g.*, BERT [5], various large scale pre-trained vision-language models [20, 26, 39, 40, 49] are proposed. The recent CLIP [32] is especially remarkable because it is trained from 400 million text-image pairs and is powerful. CLIP learns a multi-modal embedding space, which can be used to measure the semantic similarity of image and text. Using text descriptions as prompts enables CLIP the strong ability of zero-shot transfer to downstream tasks. Besides, stunning text-guided image synthesis results [7, 11, 33, 36] are enabled by CLIP through utilizing the embedding space.

3. Background

StyleCLIP [31] proposes a flexible latent mapper method for text-driven image manipulation. It is trained for a specific text command and is applicable for any image in the domain of the pre-trained StyleGAN [16]. For the text command t_{comd}, the method learns a mapper network $M_{t_{comd}}$ to yield a manipulation direction in the $\mathcal{W}+$ space given the latent image embedding $w \in \mathcal{W}+$. Then the manipulated image is obtained from a pre-trained StyleGAN generator G as $i^{'} = G(w + M_{t_{comd}}(w))$.

To train the mapper network for the purpose of achieving the text-driven manipulating effect, a CLIP loss L_C is introduced to minimize the distance between the text command t_{comd} and manipulated image $i^{'}$. L_C is formalized as:

$$\mathcal{L}_C = D_{CLIP}(i^{'}, t_{comd}), \tag{1}$$

where D_{CLIP} is the cosine distance between the embeddings of its two arguments in CLIP space. In addition, the method uses L_2 loss to norm the manipulation direction and L_{ID} loss [35] to maintain the identity of person. Hence, the overall loss is formulated as:

$$\mathcal{L}_{StyleCLIP} = \mathcal{L}_C + \lambda_{L2}\mathcal{L}_{L_2} + \lambda_{ID}\mathcal{L}_{ID}, \tag{2}$$

where λ_{L2} and λ_{ID} are the loss coefficients. Although the method can simply and efficiently achieve text-driven image manipulation without human annotations, its loss cannot distinguish between entangled and disentangled manipulations, and the manipulated results are always entangled.

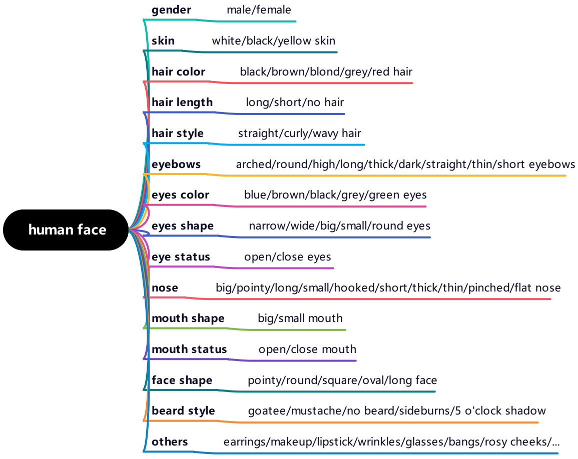

human face
gender
male/female
skin
white/black/yellow skin
hair color
black/brown/blond/grey/red hair
hair length
long/short/no hair
hair style
straight/curly/wavy hair
eyebows
arched/round/high/long/thick/dark/straight/thin/short eyebows
eyes color
blue/brown/black/grey/green eyes
eyes shape
narrow/wide/big/small/round eyes
eye status
open/close eyes
nose
big/pointy/long/small/hooked/short/thick/thin/pinched/flat nose
mouth shape
big/small mouth
mouth status
open/close mouth
face shape
pointy/round/square/oval/long face
beard style
goatee/mustache/no beard/sideburns/5 o'clock shadow
others
earrings/makeup/lipstick/wrinkles/glasses/bangs/rosy cheeks/...

Figure 2. To predict entangled attributes in face editing, we construct a hierarchical attribute structure with the help of BERT [5].

4. Predict, Prevent, and Evaluate (PPE)

The proposed PPE framework consists of three parts: 1) we design a mechanism to **Predict** the entangled attributes for given text commands; 2) based on the predicted attributes, we introduce a novel entanglement loss to **Prevent** entanglements during training, and 3) we propose a new evaluation metric to **Evaluate** disentangled text-driven image manipulation. All methods leverage the power from the large-scale pre-trained vision-language model CLIP.

4.1. Predict

We predict the entangled attributes under the assumption that entanglements result from the frequent co-occurrence of attributes in real-world images. To this end, we aggregate the images most relevant to the text command, look for the attributes that appear most frequently in the images and predict them as the entangled attributes.

Prerequisite. A predefined attribute set that includes basic visual characteristics is the prerequisite. For manipulating human faces, we need human face attributes. To obtain useful human face attributes, we firstly draw support from the large-scale pre-trained language model BERT [5]. In concrete, we let BERT predict specific attributes under different categories with designed prompts like *"a face with [MASK] eyes"*. By substituting *"eyes"* with other keywords of face characteristics, we derive various face attributes in a category-to-attribute fashion.

After further sorting and adding binary attributes like *"with earrings"*, we construct a hierarchical attribute structure (see Fig. 2) that serves for the subsequent procedures in Predict. More details are given in Appendix A.

Aggregate. This step aims to aggregate the images that are most relevant to the text command. Specifically, we propose a method based on CLIP. At first, we rank all the images in the training set *w.r.t.* their distance to the text command t_{comd} in the CLIP space. For an image $i \in I$, its ranking score is formalized as:

$$score(i) = D_{CLIP}(i, t_{comd}). \tag{3}$$

Images are ranked by their scores from small to large.

Besides, we use a zero-shot CLIP classifier to exclude the images that are classified as irrelevant in the ranked list. For single attribute manipulation, the classification labels can be obtained via a command-to-category and category-to-attributes pipeline. Take command *"blue eyes"* as an example, we firstly find its category *<eyes color>* with the help of an NLP tool (see Appendix B), then the labels $\{$*"blue eyes", "brown eyes"*$, \dots \}$ can be easily obtained via a category-to-attribute map according to the hierarchical attribute structure. Particularly, binary attributes like *"with earrings"* will trigger binary labels as $\{$*"with earrings", "without earrings"*$\}$. Afterwards, we select up to top-100 images in the left ranked list to form the command-relevant image-set $I^{'}$. The generability is discussed in Appendix C.

Find. The last step is to find the attributes that appear most frequently in the command-relevant image set $I^{'}$, except for attributes in the same category as commands. First, we rank the attributes by the sum of their CLIP distances to the images in $I^{'}$. The ranking score is formalized as:

$$score_{comd}(t_{attr}) = \sum_{i^{'} \in I^{'}} D_{CLIP}(i^{'}, t_{attr}). \tag{4}$$

Meanwhile, we consider the ranking results *w.r.t.* the full image set I. Similarly, the ranking score is:

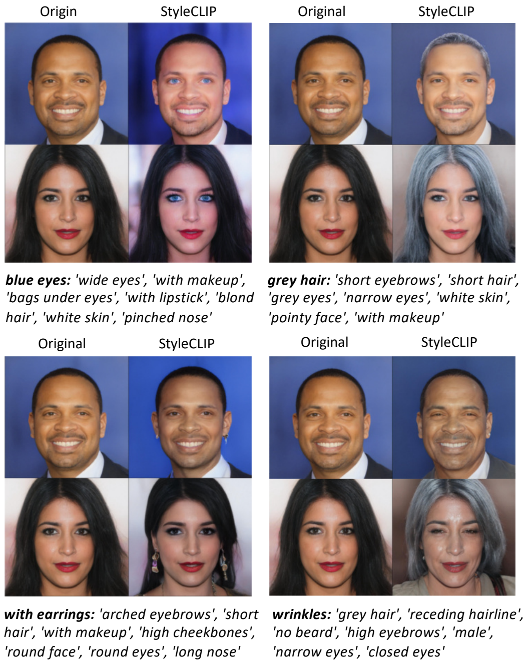
$$score_{full}(t_{attr}) = \sum_{i \in I} D_{CLIP}(i, t_{attr}). \tag{5}$$

By sorting the scores in descending order, we get r_{comd} and r_{full}. In addition, we need to find attributes that only appear frequently in the command-relevant images. For instance, *"square face"* is common for all images and thus ranks high in r_{full}, thus it may not be the entangled one even if it ranks high in r_{comd}. To this end, we adjust the r_{comd} with r_{full}. In concrete, the final ranking score is:

$$score_{final}(t_{attr}) = \frac{r_{comd}(t_{attr})}{min(r_{full}(t_{attr}), R)}, \tag{6}$$

where R is a hyper-parameter to determine if the rankings in r_{full} are high or not. Eventually, top-N attributes in the final ranked list (obtained by sorting the $score_{final}$ from small to large) are predicted as the entangled attributes $\{t_{entg^n}\}_{n=1}^{N}$.

Analysis. In Fig. 3, we illustrate some of the predictions and the corresponding manipulation results from the latent mapper of StyleCLIP. As can be seen, our method predicts the entangled attributes well, *e.g.*, *"wide eyes"* and *"with makeup"* for blue eyes, *"grey eyes"* and *"white skin"* for grey hair, *"short hair"* and *"with makeup"* for with earrings, and *"grey hair"* and *"closed eyes"* for with wrinkles.



blue eyes: *'wide eyes', 'with makeup', 'bags under eyes', 'with lipstick', 'blond hair', 'white skin', 'pinched nose'*

grey hair: *'short eyebrows', 'short hair', 'grey eyes', 'narrow eyes', 'white skin', 'pointy face', 'with makeup'*

with earrings: *'arched eyebrows', 'short hair', 'with makeup', 'high cheekbones', 'round face', 'round eyes', 'long nose'*

wrinkles: *'grey hair', 'receding hairline', 'no beard', 'high eyebrows', 'male', 'narrow eyes', 'closed eyes'*

Figure 3. Illustration of the predicted entangled attributes for various commands in text-driven image manipulation.

4.2. Prevent

For a disentangled manipulation, the command-corresponding attribute should change while other attributes should be maintained, especially for the possibly entangled ones. Therefore, based on the predicted entangled attributes $\{t_{entg^n}\}_{n=1}^{N}$, we introduce a novel entanglement loss that punishes the changes of entangled attributes after the manipulation. The changes are measured by the CLIP distances between images and texts of entangled attributes, thus the proposed entanglement loss is formulated as:

$$\mathcal{L}_E = \frac{1}{N} \sum_{n} (D_{CLIP}(i, t_{entg^n}) - D_{CLIP}(i^{'}, t_{entg^n}))^2, \tag{7}$$

where $i = G(w)$ is the original image and $i^{'}$ is the manipulated image as introduced in Sec. 3.

Together with the losses described in Eq. (2), our overall loss is defined as below:

$$\mathcal{L}_{PPE} = \mathcal{L}_C + \lambda_{L2}\mathcal{L}_{L_2} + \lambda_{ID}\mathcal{L}_{ID} + \lambda_E\mathcal{L}_E, \tag{8}$$

where λ_E is the coefficient for the entanglement loss.

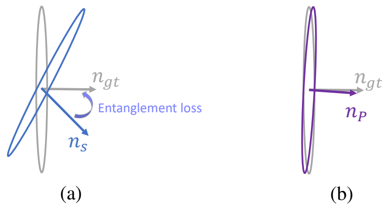


Figure 4. Illustration of (a) the effect of the entanglement loss and (b) the expected result provided by the entanglement loss.

We give an illustration in Fig. 4, where we assume there is a hyperplane in the latent space that separates having the attribute or not*, n_{gt} is the unit normal vector of the hyperplane corresponding to the command attribute, n_S is the vector found by StyleCLIP method and n_P is from PPE. As shown, the proposed entanglement loss is to constrain the model to find less entangled manipulating directions.

4.3. Evaluate

For disentangled text-driven image manipulation, we propose a new evaluation metric, *i.e.*, an indicator that evaluates the manipulation and the entanglement effects simultaneously, based on the predicted entangled attributes.

Firstly, for each text command, we quantify the manipulation effect as:

$$\triangle d_c = D_{CLIP}(i, t_{comd}) - D_{CLIP}(i^{'}, t_{comd}), \tag{9}$$

i.e., the change of command attribute in images measured by CLIP. The larger the $\triangle d_c$ is, the closer the manipulated image to the text command is in CLIP space, indicating the manipulation reaches the command-required effect.

Meanwhile, for each predicted entangled attribute, we measure the entanglement effect by:

$$\triangle d_{e^n} = D_{CLIP}(i, t_{entg^n}) - D_{CLIP}(i^{'}, t_{entg^n}), \tag{10}$$

i.e., the change of entangles attribute in images estimated by CLIP. The larger the $\triangle d_{e^n}$ is, the closer the manipulated image to the text of the entangled attribute is in CLIP space, indicating the manipulated image is entangled with the command-relevant attribute.

To reach disentangled manipulations, we expect $\triangle d_c$ to be as large as possible while $\{|\triangle d_{e^n}|\}$ to be as small as possible. Thereby, we formalize the indicator as:

$$indicator = \frac{\frac{1}{N}\sum_{n=1}^{N}|norm(\triangle d_{e^n})|}{norm(\triangle d_c)}, \tag{11}$$

where N is the number of the predicted entangled attributes and $norm(\cdot)$ is to make $\triangle d_c$ and $\{\triangle d_{e^n}\}$ comparable. In concrete, they are normalized individually as:

$$norm(\triangle d_t) = \frac{\triangle d_t}{\underset{i \in I}{max}\, D_{CLIP}(i, t) - \underset{i \in I}{min}\, D_{CLIP}(i, t)}, \tag{12}$$

where t is in $\{t_{comd}, t_{entg^1}, \ldots, t_{entg^N}\}$ and I is image set.

As described in Eq. (11), assuming $\triangle d_c$ is greater than 0 (as it should be), a high *indicator*, *e.g.*, 0.5, indicates an entangled manipulation, because when its $\triangle d_c$ increases, its $\{\triangle d_{e^n}\}$ grows correspondingly and significantly. By contrast, a lower *indicator* indicates a better disentangled manipulation, since its $\{|\triangle d_{e^n}|\}$'s changes are not as significant as its $\triangle d_c$'s. Using the *indicator*, the effect of disentangled image manipulation can be quantified.

5. Experiments

5.1. Implementation Details

To verify the proposed method, we conduct experiments on the challenging face editing task. We compare our method with our strong baseline, *i.e.*, the latent mapper in StyleCLIP [31]. Following StyleCLIP, we use the CelebA-HQ dataset [14, 25], which consists of 30,000 images, 27,176 for train-set and 2,824 for test-set; StyleGAN2 [16] pre-trained on FFHQ [15] is used to generate images; e4e [42] is used to invert images into latent embeddings in the latent space of StyleGAN2. Moreover, we train all models following the original settings as the official StyleCLIP implementation [30]. In other words, for all text commands, we train the corresponding models without tuning the hyper-parameters. We use the same loss-coefficients setting, which is $\lambda_{L2} = 0.8$ and $\lambda_{ID} = 0.1$. For the proposed entanglement loss, $\lambda_E = 100$. The number of predicted entangle attributes N in the entanglement loss (Eq. (7)) is set to 10 by default. R in Eq. (6) is set to 40, empirically.

5.2. Quantitative Results

We conduct multiple experiments using different text commands, which especially include the ones that are regarded as entangled in previous works [21, 45]. In Table 1, we illustrate the quantitative results using the evaluation metric introduced in Sec. 4.3. In concrete, *indicator* is the overall metric for disentangled image manipulation. Lower *indicator* means target manipulation is achieved with fewer entanglements, and vice versa. In addition, $\triangle d_c^{'}$ is the normalized $\triangle d_c$, and $\triangle d_e^{'}$ is the normalized $\triangle d_e$, as in Eq. (9), Eq. (10) and Eq. (12).

According to the results, we draw the following two conclusions: 1) The latent mapper method in StyleCLIP is highly entangled, and our method predicts the entangled attributes well. As can be seen in the results of "StyleCLIP", the manipulated images are closer to text commands while they are also closer to the text of entangled attributes. For example, for text command *"grey hair"* (Table 1a), when $\triangle d_c^{'}$ reaches 0.4878, $\triangle d_e^{'}$ changes at a comparable scale as *"grey eyes"* is 0.2433 and *"white skin"* is 0.2641. More significantly, for text command *"blue eyes"* (Table 1k), when $\triangle d_c^{'}$ reaches 0.4880, the $\triangle d_e^{'}$ for *"wide eyes"* is 0.3635. 2)

[*] The assumption draws from InterFaceGAN [37].

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.3359	**0.0071**
$\triangle d_c^{'}$		0.4878	0.3519
$\triangle d_e^{'}$	short eyebrows	0.1637	0.0261
	short hair	0.1945	0.0445
	with bangs	0.0927	0.0122
	grey eyes	0.2433	0.0590
	sideburns	0.1548	0.0195
	narrow eyes	0.1393	0.0126
	high cheekbones	0.0873	-0.0022
	white skin	0.2641	0.0644
	pointy face	0.1362	0.0161
	with makeup	0.1626	0.0149

(a) grey hair

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.5553	**0.1411**
$\triangle d_c^{'}$		0.3628	0.1898
$\triangle d_e^{'}$	short eyebrows	0.2362	0.0433
	with bangs	0.1583	0.0270
	short hair	0.1927	0.0245
	black eyes	0.2555	0.0458
	narrow eyes	0.2005	0.0228
	high cheekbones	0.2002	-0.0253
	with lipstick	0.1683	-0.0237
	pointy face	0.2078	0.0232
	sideburns	0.1694	0.0103
	with makeup	0.1708	0.0219

(b) black hair

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.4022	**0.1691**
$\triangle d_c^{'}$		0.2877	0.1442
$\triangle d_e^{'}$	blue eyes	0.1249	0.0235
	long hair	0.1591	0.0453
	brown hair	0.1349	0.0305
	with makeup	0.1274	0.0249
	wide eyes	0.1186	0.0246
	with earrings	0.0940	0.0165
	with bangs	0.0815	-0.0105
	pinched nose	0.1027	0.0173
	with lipstick	0.1136	0.0278
	close mouth	0.1004	0.0185

(c) wavy hair

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.3870	**0.1266**
$\triangle d_c^{'}$		0.3451	0.2384
$\triangle d_e^{'}$	short hair	0.1568	0.0525
	with lipstick	0.1249	0.0318
	smiling	0.1030	0.0155
	round eyes	0.1418	0.0242
	with makeup	0.1368	0.0221
	brown hair	0.1927	0.0552
	brown eyes	0.1316	-0.0239
	with glasses	0.0640	0.0157
	thin nose	0.1441	0.0219
	with earrings	0.1399	0.0391

(d) with bangs

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.3269	**0.1298**
$\triangle d_c^{'}$		0.3679	0.1341
$\triangle d_e^{'}$	grey hair	0.2560	0.0336
	receding hairline	0.0417	0.0035
	no beard	0.0858	0.0119
	long eyebrows	0.1132	0.0316
	long face	0.1351	0.0372
	male	0.1008	0.0035
	narrow eyes	0.1096	-0.0065
	big nose	0.0842	0.0013
	black eyes	0.0937	0.0104
	closed eyes	0.1829	0.0345

(e) with wrinkles

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.2580	**0.0900**
$\triangle d_c^{'}$		0.4072	0.3190
$\triangle d_e^{'}$	oval face	0.1498	0.0486
	small nose	0.1566	0.0524
	narrow eyes	0.1133	0.0285
	with lipstick	0.1100	0.0290
	long eyebrows	0.1245	0.0210
	short hair	0.1004	0.0303
	with bangs	0.0594	-0.0121
	receding hairline	0.0268	0.0006
	sideburns	0.0876	0.0256
	high cheekbones	0.1225	0.0404

(f) with glasses

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.4401	**0.1521**
$\triangle d_c^{'}$		0.5371	0.263
$\triangle d_e^{'}$	green eyes	0.1867	0.0270
	narrow eyes	0.3378	0.0672
	dark eyebrows	0.1920	0.0251
	with lipstick	0.1914	0.0574
	long nose	0.2805	0.0494
	high cheekbones	0.1708	0.0322
	oval face	0.3418	-0.0489
	with makeup	0.2076	0.0275
	blond hair	0.1586	0.0181
	rosy cheeks	0.2968	0.0472

(g) pale

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.3800	**0.1418**
$\triangle d_c^{'}$		0.5579	0.2452
$\triangle d_e^{'}$	open mouth	0.262	0.0376
	oval face	0.2924	0.0442
	round eyebrows	0.1990	0.0563
	big nose	0.2588	0.0243
	big mouth	0.2839	0.0485
	with lipstick	0.1156	0.0144
	sideburns	0.1311	-0.0122
	rosy cheecks	0.1566	0.0444
	closed eyes	0.2248	0.0466
	bald	0.1972	0.0218

(h) double chin

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.3069	**0.1491**
$\triangle d_c^{'}$		0.4904	0.3149
$\triangle d_e^{'}$	arched eyebrows	0.1077	0.0270
	close mouth	0.2105	0.0814
	with makeup	0.2154	0.1035
	green eyes	0.0467	0.0134
	high cheekbones	0.1517	0.0526
	oval face	0.1933	0.0482
	pinched nose	0.1070	-0.0027
	white skin	0.1668	0.0371
	big eyes	0.1157	0.0424
	rosy cheeks	0.1900	0.0611

(i) with lipstick

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.4002	**0.1881**
$\triangle d_c^{'}$		0.3585	0.1425
$\triangle d_e^{'}$	with lipstick	0.1783	0.0364
	round eyes	0.1541	0.0275
	with makeup	0.1909	0.0420
	thick nose	0.1836	0.0418
	round face	0.1222	0.0157
	rosy cheeks	0.1404	0.0121
	with earrings	0.1044	-0.0278
	double chin	0.1633	0.0410
	blond hair	0.1112	0.0243
	with bangs	0.0865	0.0175

(j) arched eyebrows

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.4220	**0.2163**
$\triangle d_c^{'}$		0.4880	0.2480
$\triangle d_e^{'}$	wide eyes	0.3635	0.1212
	with makeup	0.2127	0.0677
	bags under eyes	0.2702	0.1005
	with lipstick	0.1587	0.0350
	rosy cheeks	0.2317	0.0375
	blond hair	0.1263	0.0321
	round face	0.1993	-0.0518
	pinched nose	0.1786	0.0374
	white skin	0.2236	0.0432
	long hair	0.0949	0.0109

(k) blue eyes

		StyleCLIP	**Ours**
$indicator(\downarrow)$		0.3703	**0.2917**
$\triangle d_c^{'}$		0.4575	0.0712
$\triangle d_e^{'}$	arched eyebrows	0.1425	0.0193
	short hair	0.1369	0.0183
	with makeup	0.2054	0.0370
	high cheekbones	0.1553	0.0188
	with lipstick	0.1531	0.0210
	green eyes	0.1338	0.0115
	round face	0.1965	-0.0233
	with bangs	0.0992	0.0142
	round eyes	0.1899	0.0203
	with makeup	0.2035	0.0240

(l) with earrings

Table 1. Quantitative comparison of disentangled text-driven image manipulation with StyleCLIP [31], using the evaluation metrics introduced in Sec. 4.3. For the $indicator$, lower is better. The text command is indicated under each sub-table. Specifically, we illustrate each individual item in the changes of predicted entangled attributes $\triangle d_e^{'}$.

Our entanglement loss prevents the entanglements in image manipulation effectively. For each text command in the experiment, the $indicator$ of "Ours" is significantly lower than that of "StyleCLIP", indicating that we achieve more disentangled image manipulation. Changes on previously entangled attributes are greatly diminished (as in

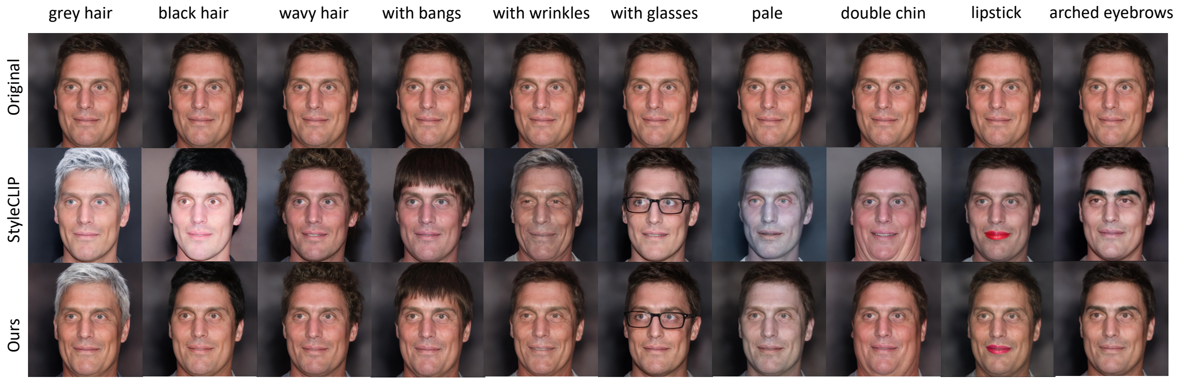
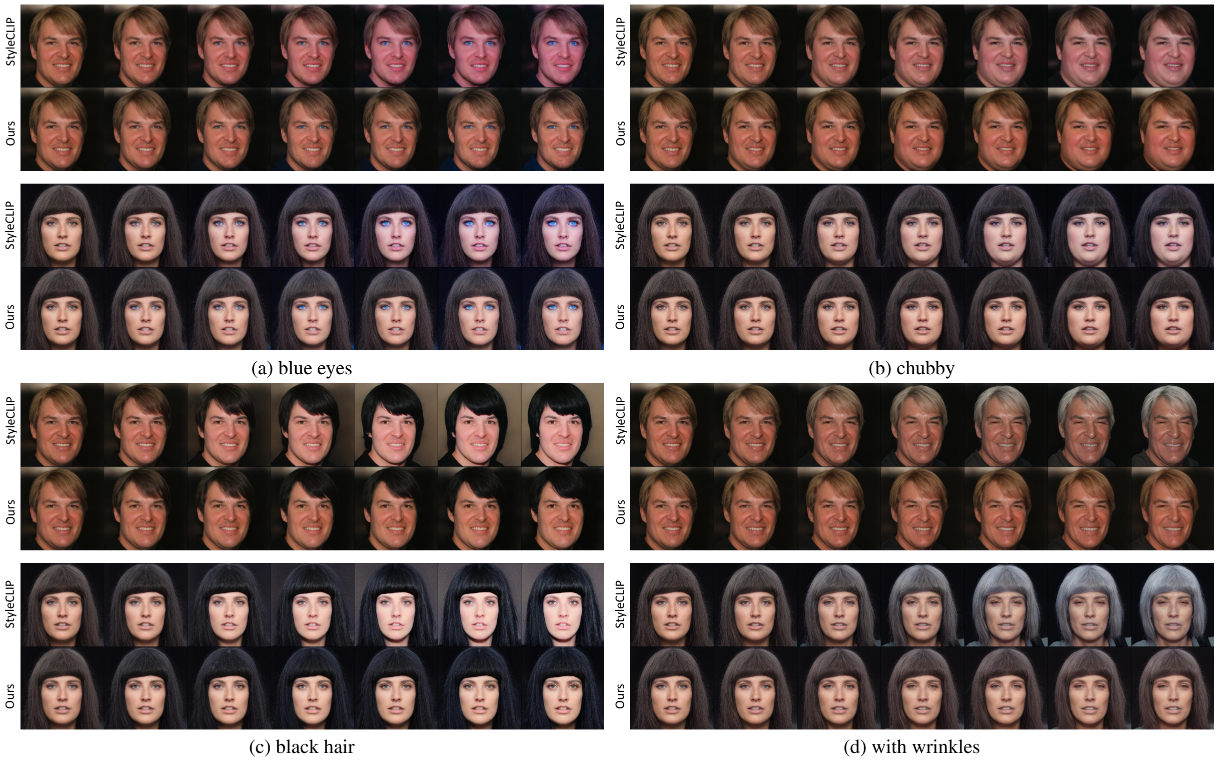


Figure 5. Qualitative comparison with StyleCLIP [31] using different text commands (indicated on the top). Ours achieves more disentangled manipulation results as only the desired attribute is manipulated while others are maintained well.



(a) blue eyes

(b) chubby

(c) black hair

(d) with wrinkles

Figure 6. Image manipulation results from StyleCLIP [31] and ours, using gradually increasing manipulation strengths. Ours present better fore-and-aft consistency along the change of manipulation strength.

$\triangle d_e^{'}$) while the manipulation effect is not affected much (according to $\triangle d_c^{'}$). In qualitative results, we illustrate that our method can achieve comparable manipulation effect with StyleCLIP when increases the manipulation strength.

5.3. Qualitative Results

Direct Manipulation Outputs. We firstly compare the directly outputted manipulation results from the trained models, without changing the manipulation strengths. In Fig. 5, we illustrate the comparing qualitative results on multiple text commands. As can be seen in the manipulation results of "StyleCLIP", it not only manipulates the required attributes, but also manipulates other attributes. Take text command *"grey hair"* as an example, the manipulated face gets grey hair, while it gets whiter skin and grey eyes simultaneously. Similarly, for the text command *"with wrinkles"*, the manipulated face gets wrinkles, grey hair, and more closed eyes in the meanwhile. Other manipulation results are obtained in similar conditions.

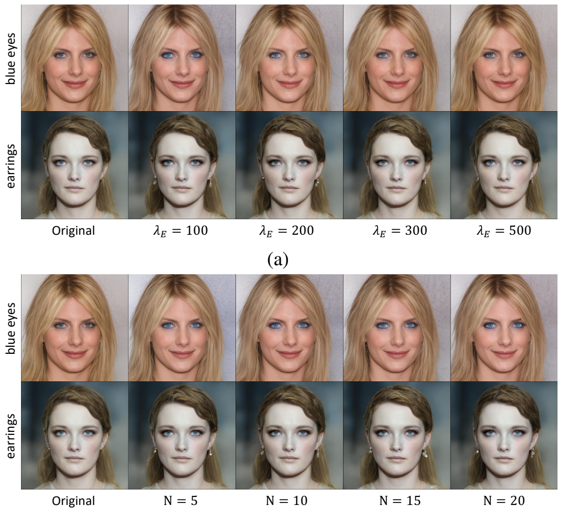
By contrast, "Ours" achieves more ideal manipulation results, where almost only the desired attribute is manipulated while other attributes of are well preserved. For example, for *"wavy hair"*, "Ours" hair becomes wavy while the hair length is close to the original one and the skin color does not become whiter; for *"double chin"*, "Ours" gets double chin while the eye color remains light brown, skin color is kept well, and mouth does not open much. In addition, it is worth mentioning that the qualitative results are quite consistent with the quantitative results, indicating that the proposed evaluation metrics are effective for the disentangled image manipulation task.

Strength-Adjusted Manipulation Outputs. We further compare the manipulation results with gradually increasing manipulation strength. To illustrate, we show four groups of comparing results in Fig. 6. In each group, we present the manipulation results for male and female, respectively. We observe that our method learns more disentangled manipulation directions compared to StyleCLIP. For StyleCLIP, when the manipulation strength increases, the desired attribute becomes more and more obvious, as well as the entangled attributes. As the male-case in Fig. 6a, from left to right, the eyes become increasingly blue while they also become wider, the face becomes whiter, and the hair color becomes lighter. Contrarily, our method presents better manipulation consistency. When the manipulation strength increases, the target attribute gradually turns apparent while others remain almost unchanged.

5.4. Discussions

Hyper-Parameters. In the previous sections, we illustrate the ability of our method to achieve disentangled image manipulation without human trials-and-errors. To further study the effects of hyper-parameters, we tune the coefficient of the proposed entanglement loss λ_E in Eq. (8) and the number of constraining attributes N in Eq. (7). We show comparing results on *"blue eyes"* and *"with earrings"*, which are found to be more entangled according to previous experimental results. As in Fig. 7a, when λ_E increases, the manipulation effects become less conspicuous while other attributes remain better. However, the manipulation effect can be enlarged by increasing the manipulation strength afterwards. As in Fig. 7b, when N varies, there are no obvious differences between the manipulation results. To conclude, our method is not sensitive to hyper-parameters.

Limitations. The limitations of the proposed PPE method are as follows: 1) Similar to StyleCLIP, the command out of the domain of CLIP and StyleGAN may not obtain ideal manipulation results. 2) The disentanglement extent in the manipulation results depends on the disentanglement extent in the latent space of StyleGAN. Since we study latent space image manipulation, the best our method can do is to find the most disentangled latent path in the latent space of pretrained generator. If the attributes are originally entangled for the generator, PPE is unable to achieve completely disentangled manipulations.



(a)

(b)

Figure 7. Hyper-parameters study.

Ethical Impact. One common issue in the image manipulation model is that it is biased toward the dataset the model was trained on. For example, BlendGAN [22] indicates that the ethical biases of a dataset may transfer to their model, *e.g*., the model outputs faces with lighter skin while the input faces are darker-skinned. Our work can help reduce this ethical impact, as our method aims at disentangled image manipulation that only changes the desired attribute while letting the others unchanged, *e.g*., our method can change the eye color while maintaining the skin color well.

6. Conclusion

We propose Predict, Prevent, and Evaluate (PPE) to achieve disentangled image manipulation with little manual effort by deeply exploiting the powerful large-scale pretrained vision-language model CLIP. CLIP is leveraged to 1) **Predict** the entangled attributes given textual manipulation command, 2) **Prevent** the model from finding entangled manipulating latent directions through a novel entanglement loss, and 3) establish a new evaluation metric that can simultaneously **Evaluate** the effects of manipulation and entanglement. PPE is tested on the challenging face editing task and is proven effective.

Acknowledgment. This work was supported by the PRIN project CREATIVE Prot. 2020ZSL9F9, by the EUREGIO project OLIVER and by the EU H2020 AI4Media project under Grant 951911.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)*, 40(3):1–21, 2021. 1, 2

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. *ACM Trans. Graph.*, 40(4), 2021. 2

[3] Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 3, 11

[6] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5706–5714, 2017. 2

[7] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, abs/2106.14843, 2021. 3

[8] Aviv Gabbay, Niv Cohen, and Yedid Hoshen. An image is worth more than a thousand words: Towards disentanglement in the wild. In *Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2

[9] Aviv Gabbay and Yedid Hoshen. Demystifying inter-class disentanglement. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[10] Aviv Gabbay and Yedid Hoshen. Scaling-up disentanglement for image translation. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[11] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021. 2, 3

[12] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020. 1, 2

[13] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020. 2

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2, 5

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2, 5

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2, 3, 5

[17] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 2

[18] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7880–7889, 2020. 2

[19] Bowen Li, Xiaojuan Qi, Philip Torr, and Thomas Lukasiewicz. Lightweight generative adversarial networks for text-guided image manipulation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22020–22031. Curran Associates, Inc., 2020. 2

[20] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020. 3

[21] Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8639–8648, 2021. 1, 2, 5

[22] Mingcong Liu, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, and Wen Zheng. Blendgan: Implicitly gan blending for arbitrary stylized face generation. In *Advances in Neural Information Processing Systems*, 2021. 8

[23] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, Nicu Sebe, and Bruno Lepri. Describe what to change: A text-guided unsupervised image-to-image translation approach. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1357–1365, 2020. 1

[24] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10785–10794, 2021. 2

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 2, 5, 11

[26] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 3

[27] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. *arXiv preprint arXiv:1810.11919*, 2018. 2

[28] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. Semi-supervised stylegan for disentanglement learning. In *International Conference on Machine Learning*, pages 7360–7369. PMLR, 2020. 2

[29] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *ACM Trans. Graph.*, 39(6), Nov. 2020. 2

[30] Or Patashnik and Zongze Wu. Official implementation of StyleCLIP. `https://github.com/orpatashnik/StyleCLIP.git`, 2021. 5

[31] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 1, 2, 3, 5, 6, 7

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 1, 2, 3

[33] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 3

[34] Facebook Research. LAMA. `https://github.com/facebookresearch/LAMA.git`, 2021. 11

[35] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2, 3

[36] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. 2, 3

[37] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 1, 2, 5

[38] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021. 1, 2

[39] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. 3

[40] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 3

[41] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020. 2

[42] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Trans. Graph.*, 40(4), July 2021. 5

[43] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020. 1, 2

[44] Binxu Wang and Carlos R Ponce. A geometric analysis of deep generative image models and its applications. In *International Conference on Learning Representations*, 2021. 1, 2

[45] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 2, 5

[46] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2256–2265, 2021. 2

[47] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Towards open-world text-guided face image generation and manipulation. *arxiv preprint arxiv: 2104.08910*, 2021. 2

[48] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13789–13798, 2021. 2

[49] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *CVPR 2021*, 2021. 3

[50] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 2